



# LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

## FAKULTAS ILMU KOMPUTER

### UNIVERSITAS BRAWIJAYA

---

BAB : PENGENALAN OBJECT ORIENTED PROGRAMMING  
NAMA : NIAN DINI ARTI  
NIM : 155150200111051  
TANGGAL : 23/09/2016  
ASISTEN : RIZKI MAULANA AKBAR

---

#### A. DEFINISI MASALAH

1. Buatlah program untuk membuat kalkulator penjumlahan, pengurangan, perkalian dan pembagian dengan menggunakan parameter dan argument

#### B. SOURCE CODE

**Class Calculate :**

```
1 public class Calculate {
2     private double x;
3     private double y;
4     private double hasil;
5
6     public Calculate(double x) {
7         this.x = x;
8         this.hasil = x;
9     }
10
11     public double jumlah(double y) {
12         this.hasil = this.x + y;
13         return this.hasil;
14     }
15
16     public double kurang(double y) {
17         this.hasil = this.hasil - y;
18         return this.hasil;
19     }
20
21     public double kali(double y) {
22         this.hasil = this.hasil * y;
23         return this.hasil;
24     }
25
26     public double bagi(double y) {
27         this.hasil = this.hasil / y;
28         return this.hasil;
29     }
30
31     public double hasil() {
32         return this.hasil;
33     }
34 }
```

### Class Main untuk Calculate :

1	import java.util.Scanner;
2	public class MainCalculate {
3	public static void main(String[] args) {
4	Scanner in = new Scanner(System.in);
5	System.out.println("Masukkan angka awal : ");
6	double awal = in.nextDouble();
7	String n;
8	Calculate a = new Calculate(awal);
9	System.out.println("Operator (/,*,+,-,=) : ");
10	n = in.next();
11	while (!"=".equals(n)) {
12	if (n.equals("-")) {
13	awal = in.nextDouble();
14	a.kurang(awal);
15	System.out.println("Operator (/,*,+,-,=) : ");
16	n = in.next();
17	} else if (n.equals("+")) {
18	awal = in.nextDouble();
19	a.jumlah(awal);
20	System.out.println("Operator (/,*,+,-,=) : ");
21	n = in.next();
22	} else if (n.equals("/")) {
23	awal = in.nextDouble();
24	a.bagi(awal);
25	System.out.println("Operator (/,*,+,-,=) : ");
26	n = in.next();
27	} else if (n.equals("*")) {
28	awal = in.nextDouble();
29	a.kali(awal);
30	System.out.println("Operator (/,*,+,-,=) : ");
31	n = in.next();
32	} else {
33	System.out.println("Operasi yang anda masukkan
34	tidak tersedia");
35	break;
36	}
37	System.out.println("Hasil : ");
38	System.out.println(a.hasil());
39	}
40	}

## C. PEMBAHASAN

### Class Calculate :

1	Deklarasi kelas Calculate dengan tipe public
2	Deklarasi atribut x bertipe data double
3	Deklarasi atribut y bertipe data double
4	Deklarasi atribut hasil bertipe data double
5	Deklarasi konstruktor dengan parameter x bertipe data double
6	Dengan argumen nilai variabel x milik kelas Calculate sama dengan nilai variabel x

	pada parameter
7	Dan nilai variabel hasil milik kelas Calculate sama dengan nilai variabel x pada parameter
8	Akhir dari konstruktor
9	Deklarasi method public jumlah bertipe double dengan parameter variabel y bertipe double
10	Inisialisasi nilai variabel hasil milik kelas dengan nilai variabel hasil kelas ditambah nilai variabel y
11	Mengembalikan nilai variabel hasil milik kelas
12	Akhir method jumlah
13	Deklarasi method public kurang bertipe double dengan parameter variabel y bertipe double
14	Inisialisasi nilai variabel hasil milik kelas dengan nilai variabel hasil kelas dikurang nilai variabel y
15	Mengembalikan nilai variabel hasil milik kelas
16	Akhir method kurang
17	Deklarasi method public kali bertipe double dengan parameter variabel y bertipe double
18	Inisialisasi nilai variabel hasil milik kelas dengan nilai variabel hasil kelas dikalikan nilai variabel y
19	Mengembalikan nilai variabel hasil milik kelas
20	Akhir method kali
21	Deklarasi method public bagi bertipe double dengan parameter variabel y bertipe double
22	Inisialisasi nilai variabel hasil milik kelas dengan nilai variabel hasil kelas dibagi nilai variabel y
23	Mengembalikan nilai variabel hasil milik kelas
24	Akhir method bagi
25	Deklarasi method public hasil bertipe double
26	Mengembalikan nilai variabel hasil milik kelas
27	Akhir method
28	Akhir kelas.

#### **Class Main untuk Calculate :**

1	Megimport class Scanner
2	Deklarasi class MainCalculate
3	Deklarasi method main
4	Memanggil kelas Scanner
5	Mencetak "Masukkan angka awal : "
6	Menginput nilai variabel awal bertipe double
7	Deklarasi variabel string n
8	Memanggil kelas Calculate
9	Mencetak "Operator (/,*,+,-,=) : "
10	Menginput nilai variabel n
11	Proses seleksi while dengan syarat apabila nilai inputan tidak sama dengan "=" maka statement akan dijalankan
12	Proses seleksi if dengan syarat apabila inputan bernilai sama dengan "-" maka,
13	Memasukkan nilai variabel awal
14	Memanggil method kurang dengan parameter variabel awal menggunakan objek a
15	Mencetak "Operator (/,*,+,-,=) : "

16	Memasukkan nilai variabel n
17	Proses seleksi else if dengan syarat apabila inputan bernilai sama dengan “+” maka,
18	Memasukkan nilai variabel awal
19	Memanggil method jumlah dengan parameter variabel awal menggunakan objek a
20	Mencetak “Operator (/,*,+,-,=) : ”
21	Memasukkan nilai variabel n
22	Proses seleksi else if dengan syarat apabila inputan bernilai sama dengan “/” maka,
23	Memasukkan nilai variabel awal
24	Memanggil method bagi dengan parameter variabel awal menggunakan objek a
25	Mencetak “Operator (/,*,+,-,=) : ”
26	Memasukkan nilai variabel n
27	Proses seleksi else if dengan syarat apabila inputan bernilai sama dengan “*” maka,
28	Memasukkan nilai variabel awal
29	Memanggil method kali dengan parameter variabel awal menggunakan objek a
30	Mencetak “Operator (/,*,+,-,=) : ”
31	Memasukkan nilai variabel n
32	Proses seleksi else
33	Mencetak “Operasi yang anda masukkan tidak tersedia”
34	Menghentikan perulangan
35	Akhir dari else
36	Akhir dari while
37	Mencetak “Hasil : “
38	Mencetak nilai yang ada pada method hasil melalui objek a
39	Akhir method main
40	Akhir class

#### D. SCREENSHOT PROGRAM

```

run:
Masukkan angka awal :
144
Operator (/,*,+,-,=) :
/
2
Operator (/,*,+,-,=) :
*
3
Operator (/,*,+,-,=) :
+
5
Operator (/,*,+,-,=) :
-
2
Operator (/,*,+,-,=) :
=
Hasil :
147.0
BUILD SUCCESSFUL (total time: 33 s)

```

#### E. KESIMPULAN

Object Oriented Programming (OOP) adalah metode yang digunakan

untuk mendapatkan solusi dari suatu masalah melalui prespektif objek. Objek dapat didefinisikan sebagai entitas yang memiliki data dan method. Pemrograman berorientasi objek memiliki beberapa perbedaan dengan pemrograman terstruktur, diantaranya :

1. Pada pemrograman prosedural masalah diselesaikan dalam bentuk prosedur atau fungsi, sedangkan pada OOP fungsi dan data menjadi satu kesatuan yang disebut objek.
2. Pada pemrograman prosedural program merupakan urutan instruksi, sedangkan pada OOP program bukan berupa urutan instruksi tetapi merupakan beberapa objek yang bekerjasama untuk menyelesaikan masalah.
3. Pada pemrograman prosedural data bersifat pasif sedang pada OOP objek bersifat aktif.

Dalam pemrograman berorientasi objek terdapat method void dan non void. Perbedaan dari kedua method ini adalah method void tidak memiliki return value sedangkan method non void memiliki return value. Contohnya :

Method void :

	<pre>public void sisipData(int dt){     Node nodeBaru;     nodeBaru = new Node(dt, pHead);     pHead = nodeBaru; }</pre>
--	--

Method non void :

	<pre>public double jumlah(double y) {     this.hasil = this.x + y;     return this.hasil; }</pre>
--	---

Sedangkan yang dimaksud dengan return value adalah nilai yang dikembalikan oleh suatu method non void. Contohnya :

	<pre>public double bagi(double y) {     this.hasil = this.hasil / y;     return this.hasil; }</pre>
--	---